

Desarrollo de entornos virtuales aplicados a dispositivos hápticos

GUDIÑO-LAU, Jorge*†, LINARES-RAMIREZ, Enrique, CHARRE, Saida y DURAN, Miguel.

Universidad de Colima, Av. Universidad 333, Las Víboras, 28040 Colima, Col., México

Recibido Octubre 14, 2016; Aceptado Noviembre 25, 2016

Resumen

En este artículo se muestra el desarrollo de un entorno virtual y presentan conceptos básicos de renderizado, manejo de archivos, computación gráfica, háptica de computadora entre otros. El proyecto presenta de forma sencilla un algoritmo de detección de colisiones en fase amplia que sirve como base del manejo del robot manipulador e inicio del control háptico. Todo desarrollado en el software de c++, ya que es muy sencillo de manejar y adquirir. El objetivo de este trabajo es mostrar aplicaciones del área de la háptica, al desarrollar entornos virtuales en 3D, empleando el software más popular en el área de sistemas computacionales como es c++ y con la ayuda de OpenGL.

Háptica, entorno virtual, colisiones, software, C++, OpenGL

Abstract

This paper describe the development of a virtual environment and present basic concepts of rendering, file management, computer graphics, computer haptics among others are shown. The project present a simply an algorithm of collision detection in wide phase that serves as base management of manipulator robot and the begining of the haptic control. All is developed in software c ++, because it is very easy to use and acquire. The object of this project is to show applications of the haptic area, to develop 3D virtual environments using the most popular software in the area of computer systems such as c ++ and the help of the OpenGL.

Haptics, virtual environment, collision, c++ software, OpenGL

Citación: GUDIÑO-LAU, Jorge, LINARES-RAMIREZ, Enrique, CHARRE, Saida y DURAN, Miguel. Desarrollo de entornos virtuales aplicados a dispositivos hápticos. Revista de Sistemas Computacionales y TIC'S 2016, 2-6: 124-131

* Correspondencia al Autor (Correo Electrónico: jglau@uclm.mx)

† Investigador contribuyendo como primer autor.

Introducción

Los avances en la tecnología se dan cada vez más rápido, las computadoras están presentes prácticamente en todos lados, desde un horno de microondas hasta los teléfonos celulares. Sin las computadoras sería prácticamente imposible llevar el tipo de vida que llevamos, esto las hace una herramienta imprescindible. Es por esto que los alumnos de las ingenierías requieren los conocimientos para la solución de problemas aplicando la tecnología actual.

Estos problemas se presentan en todas las áreas de especialidad, lo que conlleva a otra interrogante: ¿Es necesario preparar profesionistas que sean expertos en todo?, se ha demostrado que la especialización es actualmente la mejor forma de sobresalir y de realizar productos novedosos, de calidad y sobre todo que resuelvan problemas importantes. Pero, si muchos de los problemas que se desean resolver son multidisciplinarios, lo que se requiere es que especialistas de diferentes áreas analicen el problema desde diferentes ángulos para después proponer una solución conjunta.

Por un lado se tiene la necesidad de desarrollar software, sin el cual las computadoras son prácticamente inútiles, por otro lado, las computadoras son dispositivos electrónicos, al igual que muchos dispositivos utilizados actualmente – como los teléfonos celulares. Si a esto se le agrega que existen vehículos mecánicos a los cuales se les ha adaptado tecnología electrónica de cómputo, por ejemplo los robots que cada vez están más relacionados con la vida diaria, existen problemas que pueden englobar a los estudiantes de muchas disciplinas de la ingeniería:

- Sistemas Computacionales
- Tecnologías Electrónicas
- Mecánica
- Mecatrónica

De todos es bien conocido que cualquier rama de la ingeniería parte del diseño de los componentes/herramientas antes de buscar la solución de algún problema. Es en esta área donde se pueden aplicar los conocimientos adquiridos en las Facultades para diseñar y desarrollar herramientas que permitan una mejor comprensión de las aplicaciones que tienen los conocimientos adquiridos.

Una herramienta que se ha vuelto fundamental en muchas áreas es la llamada Entorno Virtual (VE). Los entornos virtuales son programas de computadora con imágenes interactivas en tres dimensiones (3D) que permiten analizar situaciones reales en un entorno no real. Esto lo convierte en una excelente herramienta para el análisis previo a la construcción, de esta manera se pueden detectar errores en el diseño o realizar ajustes previos con lo que se ahorra una cantidad de dinero considerable.

La háptica se refiere al sentido del tacto con fines de manipulación de objetos. El sentido del tacto es el único que presenta un canal bidireccional de comunicación con el mundo exterior (Srinivasan, 2004), no solo se pueden percibir los estímulos como el toque de una persona, también permite transmitirlos al entorno. El estudio de esta disciplina aplicado a la interacción hombre-máquina es relativamente joven comparado con otras áreas de investigación (Figura 1).

Debido al ámbito multidisciplinario de aplicación de la háptica, se han creado 3 áreas principales (Srinivasan, 2004):

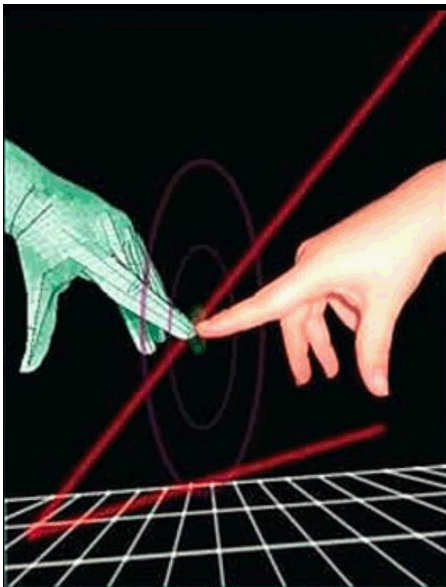


Figura 1 Interacción hombre-máquina (Otađu, Ling, 2008)

- **Háptica humana.** Es el estudio de la percepción y manipulación de objetos a través del tacto.
- **Háptica de máquina.** Abarca el diseño, construcción y uso de las máquinas para reemplazar o mejorar el sentido del tacto.
- **Háptica de computadora.** Conjunto de algoritmos y software asociado con la generación del contacto y percepción de objetos virtuales

Se han desarrollado una gran cantidad de dispositivos hápticos que permiten la interacción hombre-máquina. El tipo y las características de cada uno dependen de la aplicación para la que han sido diseñados. Tal es el caso del Novint Falcon (Figura 2), un dispositivo háptico con tres grados de libertad (3DOF) el cual permite movimiento en un espacio tridimensional y además permite transmitir fuerza al usuario, lo que da la apariencia de estar tocando realmente un objeto.



Figura 2 Dispositivo háptico Falcon de Novint. (Novint.com)

Desarrollo

En la actualidad, el uso de herramientas como los entornos virtuales representan un gran apoyo en el diseño y análisis sin embargo esta información es meramente visual, eso significa que se puede interactuar con el entorno a través del mouse de la computadora (tradicionalmente) pero existen situaciones en las que este tipo de interactividad no es suficiente para el análisis realista de la escena. Por ejemplo, si se tiene el diseño en 3D de un objeto que se desea construir y se desea conocer como será su textura al tacto, esto es algo que con el uso del mouse sería imposible. Es aquí donde entra el uso de otro tipo de tecnología que hace algunos años podría haber parecido inviable para estas situaciones, la háptica.

El presente trabajo presenta las bases del desarrollo de un entorno virtual y los principios básicos del funcionamiento del Novint Falcon para uso en el entorno.

El proyecto se diseñó con una idea en mente: Enseñar a los alumnos los principios fundamentales del desarrollo de un entorno virtual a través de la manipulación de un dispositivo háptico, el cual debe tener la particularidad de que su código fuente sea portable a varios Sistemas Operativos (OS).

Una de las principales premisas es que el código fuente sea portable, esto significa que el mismo código fuente se pueda compilar tanto en Windows, Mac, Linux, etc. sin realizar ningún cambio en él. Esto requirió de investigar las librerías para el manejo tanto de gráficos como del Falcon.

En cuanto a las librerías gráficas se encontraron dos principales que son las más utilizadas: OpenGL y DirectX (Stanney, Zyda, 2002). Ambas con una gran calidad en la visualización de gráficos, ambas cuentan con muchos videojuegos y aplicaciones desarrolladas con ayuda de estas. De las dos, la única que proporciona soporte para diversos sistemas es OpenGL, por lo que se decidió utilizarla para el proyecto (DirectX es diseñada por Microsoft para OS Windows). Para el manejo del Falcon, se encontraron varias librerías multiplataforma, entre ellas se tiene CHAI3D y libnifalcon. Se eligió libnifalcon porque es ligera y además se pueden desarrollar aplicaciones hápticas sin necesidad de una interfaz gráfica, como el CHAI3D que contiene muchas herramientas gráficas para la creación de objetos (Kadlecek, 2010), lo cual sirve para otro de los objetivos del proyecto: Diseño de la interfaz gráfica desde cero, es decir, sin herramientas que tengan prediseñados los objetos.

Como lenguaje de programación se escogió el C++, además de ser el más utilizado en diferentes áreas, es perfecto para la enseñanza de la programación porque permite desarrollar aplicaciones de todo tipo.

Para iniciar con el diseño de un entorno virtual se deben conocer los fundamentos de las gráficas por computadora (CG), posteriormente o a la par, se requiere del conocimiento básico de ciertos aspectos matemáticos como vectores, matrices, etc. que a pesar de estar agrupados en la industria bajo el concepto de Matemáticas para CG, académicamente se dividen en Álgebra Lineal, Geometría, etc.

En el lado de la programación, es necesario que se haya cursado al menos un semestre de la materia programación, algorítmica, fundamentos de programación o alguna similar, también es necesario el conocimiento del lenguaje de programación C++ (en algunas asignaturas mencionadas anteriormente se lleva a la par). El diseño de algoritmos es fundamental para el desarrollo de un VE.

Es necesaria la utilización de ciertas librerías (además de las ya mencionadas) para el desarrollo de los VE, de entre todas las existentes se eligieron freeglut y glew, las cuales están ampliamente documentadas en la web.

Debido a la naturaleza multiplataforma que se desea para el software, resulta indispensable el manejo de la programación de la interfaz en los distintos OS, esto requiere de un gran esfuerzo de programación para desarrollar código que genere las ventanas y los elementos requeridos para cada uno de ellos. Pensando en esto, se eligió la librería freeglut que es multiplataforma y encapsula la creación de ventanas en múltiples, lo que libera al programador de la tediosa tarea de preocuparse de ello.

OpenGL viene preinstalado en todas las computadoras, el fabricante de las tarjetas de video diseña las librerías para el OS en particular sobre el que se instalarán los drivers de video. Normalmente la programación de OpenGL se realiza a través de librerías de terceros que agrupan las funciones y tipos de datos y facilitan el diseño de programas. Una de estas librerías es glew. Ambas librerías deben ser compiladas para el OS con el que se esté trabajando, las instrucciones se incluyen con el código fuente.

Para el dibujado de las figuras se requiere de una gran cantidad de código ya que se tiene que codificar la posición de cada vértice en coordenadas del mundo para ir creando los vértices de los polígonos. A fin de extender los conocimientos necesarios para el diseño de la aplicación y pensando en tener la menor cantidad de código posible, se eligió utilizar un programa de edición 3D como el Blender para diseñar los objetos 3D, posteriormente se exportan al formato de archivo .obj (de Wavefront) el cual contiene la descripción de la geometría del objeto.

El solo hecho de exportar el modelo a un archivo con formato obj no es suficiente para que se pueda visualizar el objeto en la aplicación, se requiere el diseño de una librería que convierta la información del archivo a información de vértices que pueda ser utilizada para el dibujo de los polígonos. En el desarrollo de esta librería se requieren conocimientos básicos de manejo de archivos, punteros y de estructuras de datos.

Se utilizan las listas enlazadas (Figura 3) como estructura de datos para promover el uso de técnicas computacionales. Las listas enlazadas son, como su nombre lo indica listas de nodos divididos en dos partes: una para la información y otra que se utiliza como enlace (apuntador) al nodo siguiente. Este enlace al nodo siguiente permite que la estructura crezca tanto como se requiera, esto permite que sea dinámica.



Figura 3 Representación de una lista enlazada

Dentro de la definición del archivo .obj se tiene una gran cantidad de información, de la cual solo nos interesa (por el momento) la información de los vértices y de las caras (conjuntos de vértices) que forman la imagen. Para poder representar mejor los dos tipos de información del archivo se crean dos tipos de listas diferentes.

Una lista enlazada simple para el almacenamiento de la información de los vértices (valores x,y,z). También se crea una lista enlazada en varios niveles (a manera de tabla) para almacenar las caras del objeto y la información de los vértices que pertenecen a cada cara. El esquema mencionado proporciona flexibilidad para abrir distintos archivos sin que se presente algún error.

Existen actualmente diversos programas de edición en 3D: Maya, Blender, 3DStudio, etc. De entre todos se eligió Blender por ser gratuito y multiplataforma además de ser sencillo de utilizar. En la web se tiene una gran cantidad de manuales de Blender por lo que no se ahondará en el uso de la aplicación. Lo que si se mencionará es que una vez diseñado el objeto en Blender, este debe ser exportado al formato de archivo obj para que el entorno sea capaz de leerlo (Figura 4).

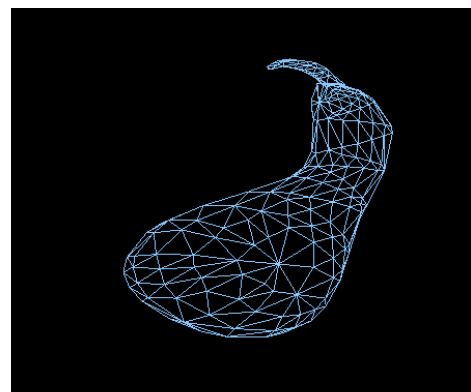


Figura 4 Modelo 3D renderizado como una malla a partir de un archivo .obj

En un entorno 3D que se visualice en una computadora se utiliza normalmente el mouse como medio de exploración, esto tiene una pequeña limitante, el mouse solo se mueve en 2D. Esta es una de las razones por las que se eligió utilizar un dispositivo háptico para la exploración del entorno.

Se eligió en Novint Falcon, el cual, gracias a sus 3DOF genera movimiento en 3D. Por otro lado se eligió un dispositivo háptico por permitir transmitir al usuario fuerza, la cual al combinarse con las imágenes mostradas en el entorno generan la sensación en el usuario de estar tocando al objeto.

Esta sensación de tocar al objeto por medio de la retroalimentación de fuerzas generada por el dispositivo háptico debe ser programada también, lo que presenta un excelente reto para los estudiantes al instalar las librerías necesarias para su manipulación. La librería elegida viene en código fuente por lo que debe ser compilada específicamente para el OS anfitrión, las instrucciones de compilación se proporcionan en la página del desarrollador.

La estimulación sensorial de tocar un objeto relacionadas. Por un lado está la parte visual que genera el entorno, en esta área se cuenta con algoritmos que permiten detectar la colisión entre objetos y en consecuencia elegir la acción que se debe tomar. Por otro lado está la parte háptica que se logra por medio de valores que representan la fuerza que debe generar el dispositivo.

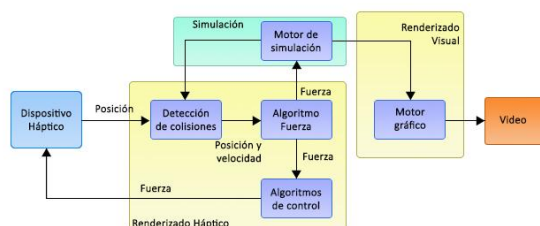


Figura 5 Ciclo háptico (Salisbury, Conti, 2004)

La detección de colisiones es un aspecto que se considera esencial en todo VE (Figura 5), desde los videojuegos hasta los costosos simuladores de vuelo. De los algoritmos con que se cuenta para detectar las colisiones se eligió el más sencillo para adentrar a los alumnos en este campo. El algoritmo es el llamado detección en fase amplia (Erlben, Sporning, Henriksen, et. al, 2012).

La detección de colisiones en fase amplia se basa en construir una caja alrededor del objeto. Dicha caja tiene sus caras perpendiculares a los ejes de coordenadas mundiales.

Una vez obtenida esta la caja, se proyecta la longitud de cada uno de sus caras sobre los ejes correspondientes como se puede observar en la Figura 6. Este mismo proceso se lleva a cabo para el Haptic Interface Point (HIP) que es una representación del puntero (Kawai, Faust, et. al, 2012). Al hacer las proyecciones de las caras sobre los ejes lo que se logra es reducir la detección de colisiones a 3 simples comparaciones: si se sobreponen las líneas del HIP y del objeto en los 3 ejes se tiene entonces una colisión (Figura 7).

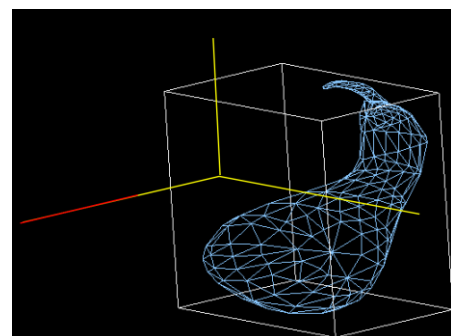


Figura 6 Proyección de la caja de colisión en uno de los ejes (en rojo).

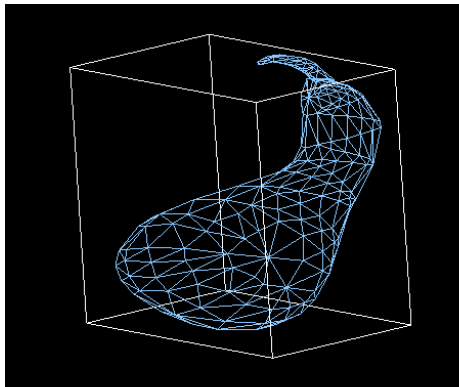


Figura 7 Modelo 3D delimitado por la caja utilizada para la detección de colisiones en fase amplia

Al detectarse una colisión se debe notificar al dispositivo háptico que represente o genere una fuerza tal, que el usuario perciba estar tocando el objeto. Para el cálculo de la fuerza a generar por el dispositivo se utiliza el método del resorte, el cual es sencillo de aplicar. Este método se basa en la premisa que en un VE un objeto siempre puede ser penetrado, no importa la dureza que se le desee dar a la superficie (Figura 8a).

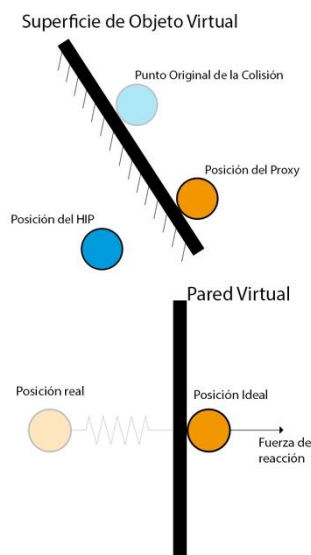


Figura 8 Lado izquierdo (a) se muestra la representación virtual de una colisión. En el lado derecho (b) se muestra la aplicación de un resorte virtual para calcular la fuerza a enviar al dispositivo háptico

Visualmente el puntero, al cual se da el nombre de proxy, no penetra el objeto virtual, internamente el proxy si lo hace, con lo que se genera una diferencia de posición entre ambos, esta diferencia de posición es la que se utiliza para calcular la fuerza a representar por el dispositivo háptico. Para poder representar correctamente la fuerza, se tiene un resorte imaginario atado a caja que rodea el objeto (ver Fig. 8b). Una vez es penetrado el mismo, se toma la distancia entre el proxy y el HIP como distancia de elongación del resorte, aplicando la ley de Hooke se puede entonces obtener un valor de fuerza ejercida. Ese valor es el que se envía al dispositivo háptico para generar las fuerzas.

Resultados

Los alumnos que participaron en el diseño de un entorno virtual como el presentado mostraron mucho interés en continuar desarrollando aplicaciones similares o mejorando la presentada. Analizan la razón de la existencia de muchas materias del plan de estudios que pensaban no tenían utilidad práctica, al desarrollar aplicaciones como la presentada, valoran los conocimientos adquiridos en clase y se interesan más en su área de estudios.

Conclusiones

El uso de proyectos multidisciplinarios es de gran utilidad al momento de evaluar los conocimientos adquiridos, incluso para motivar al alumno a continuar investigando lo cual se refleja no solo en una mejora de las calificaciones, si no, también en una mejor calidad del profesionalista al egresar.

En la Facultad de Ingeniería Electromecánica se tiene este tipo de proyectos denominados “proyectos integradores” con los cuales se pretende combinar los conocimientos adquiridos en las materias cursadas en la solución de un problema.

Se dieron los primeros pasos en el área de las interfaces hápticas, se diseñó con el objetivo de diagnosticar y evaluar pacientes con algún daño en extremidades superiores, en manos específicamente. Todavía no se aplica a pacientes con, el siguiente paso es realizarlo con niños que presenten algún padecimiento. La realidad virtual y la retroalimentación de fuerzas son muy útiles al momento de realizar seguimiento de trayectorias dirigidas, ya que ofrece un mejor control sobre la ruta que se plantea seguir, y de esta forma se puede diagnosticar y evaluar a un paciente; ya que el algoritmo de control de fuerza utilizado es bastante sencillo permite representar fuerzas de una forma aceptable.

Referencias

Erleben, Kenny. Sporning, Jon. Henriksen, Knud, et. al. (2012) Física para Videojuegos. México: Cengage Learning.

K. Salisbury, F. Conti, Francois. "Haptic Rendering: Introductory Concepts". Computer Graphics and Applications, IEEE, 2. 24. 2004. 24-32.

Kadlecek, Petr (2010). A Practical Survey of Haptic APIs. Tesis de Licenciatura, Faculty of Mathematics and Physics, Charles University in Prague, Praga.

Kawai, Satoru, Faust Jr., Paul H. y MacKenzie, Christine L. (2012). "Computer Graphic and PHANToM Haptic Displays: Powerful Tools in Understand How Human Perceive Heaviness". En: El Saddik, Abdulmotaleb (ed.), Haptics Rendering and Applications (pp. 25-46). Rijeka, Croatia: InTech.

Otaduy, Miguel A. Lin, Ming C. (2008). Haptic rendering. Foundations, algorithms, and applications. India: A. K. Peters, Ltd.

Srinivasan, Mandayam A. (2004). What is Haptics?. Recuperado de <http://touchlab.mit.edu>

Stanney, Kay M., Zyda, Michael (2002). Virtual Environments in the 21st Century. En: Stanney, Kay (ed.), Handbook of virtual environments. Design, Implementation, and Applications (pp. 1-14). Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc. Publishers.